

```

1 // [木塊問題/The Blocks Problem]
2 #define IN "P0505IN.txt"
3 #define OUT "P0505OUT.txt"
4 //*****
5 #include <iostream>
6 #include <time.h>
7 using namespace std;
8 void redir(void);
9 //*****
10 /* Work Space*/
11 #include <string>
12 #include <vector>
13
14 void find_block(int a, int &pa, int &ha);
15 void clear_above(int p, int h);
16 void pipe_over(int p, int h, int p2);
17 void show(void);
18
19 int n;
20 vector<int> pipe[25]; //二維陣列(第一維大小根據題意, 第二維大小不固定)
21 //*****
22 int main(void)
23 {
24     redir(); //redirection
25 //*****
26 /* Work Space*/
27     int i;
28     string s1, s2;
29     int a, b;
30     int pa, pb; //pa, pb: position of a and b
31     int ha, hb; //ha, hb: height of a and b
32
33     cin >> n;
34     for(i=0; i<n; i++){
35         pipe[i].push_back(i);
36     }
37
38     while(1){
39         cin >> s1;
40         if(s1 == "quit"){
41             break;
42         }else{
43             cin >> a >> s2 >> b;
44
45             find_block(a, pa, ha);
46             find_block(b, pb, hb);
47
48             if(pa == pb){
49                 continue; //非法指令
50             }
51             if(s1 == "move"){
52                 clear_above(pa, ha);
53             }
54
55             if(s2 == "onto"){
56                 clear_above(pb, hb);
57             }
58
59             pipe_over(pa, ha, pb);
60         }
61     }
62     show();
63 //*****

```

```

64     freopen("CON", "r", stdin); //取消重新導向
65     freopen("CON", "w", stdout);
66
67     printf("Time used = %.2f\n", (double)clock()/CLK_TCK); //傳回程式目前為止執行的時間
68
69     system("pause");
70     return 0; //the end...
71 }
72
73 void redir(void)
74 {
75     freopen(IN, "r", stdin);
76     freopen(OUT, "w", stdout);
77 }
78 //*****
79 /* Work Space*/
80 void find_block(int a, int &p, int &h)
81 {
82     for(p=0; p<n; p++){
83         for(h=0; h<pipe[p].size(); h++){
84             if(pipe[p][h] == a){
85                 return;
86             }
87         }
88     }
89 }
90
91 //把第p堆高度為h的木塊"上方"的所有木塊移回原位
92 void clear_above(int p, int h)
93 {
94     int i;
95     int b;
96
97     for(i=pipe[p].size()-1; i>h; i--){
98         b = pipe[p][i];
99         pipe[b].push_back(b);
100     }
101     pipe[p].resize(h+1);
102 }
103
104 //把第p堆高度h"及其上方"的木塊整體移動到p2堆的頂部
105 void pipe_over(int p, int h, int p2)
106 {
107     int i;
108
109     for(i=h; i<pipe[p].size(); i++){
110         pipe[p2].push_back(pipe[p][i]);
111     }
112     pipe[p].resize(h);
113 }
114
115 //輸出結果
116 void show(void)
117 {
118     int i, j;
119
120     for(i=0; i<n; i++){
121         printf("%d: ", i);
122         for(j=0; j<pipe[i].size(); j++){
123             printf("%d ", pipe[i][j]);
124         }
125         printf("\n");
126     }

```

```
127 }
128
129 //Input(IN) Sample
130 /*
131 10
132 move 9 onto 1
133 move 8 over 1
134 move 7 over 1
135 move 6 over 1
136 pile 8 over 6
137 pile 8 over 5
138 move 2 over 1
139 move 4 over 9
140 quit
141 */
```